I V A N Users Manual

(Version 1.0)

by

Sanford Weisberg
and
Kenneth J. Koehler

Technical Report #266

15 October 1976

University of Minnesota
School of Statistics

This work is supported by grants from the University of Minnesota Educational Development Program.

Version 1.0
October, 1976

IVAN USER'S MANUAL

by
Sanford Weisberg
and
Kenneth J. Koehler

University of Minnesota
Department of Applied Statistics

0. Introduction
1. Data files, file types, and program limitations (p. 3)
2. Specifying a model (p. 7)
3. Commands
   - 3.1 MODEL: Specify a new model (p. 14)
   - 3.2 ANOVA: Print analysis of variance table (p. 15)
   - 3.3 FTEST: Perform F tests (p. 16)
   - 3.4 CONTRASTS: Arbitrary linear contrasts (p. 17)
   - 3.5 POLY: Compute and print polynomial contrasts (p. 20)
   - 3.6 MEANS: Print means (p. 22)
   - 3.7 RESIDUALS: Print residuals (p. 24)
   - 3.8 TRAN: Transform the data (p. 25)
   - 3.9 CHANGE: Modify an observation (p. 26)
   - 3.10 ERRORSS: Print residual sum of squares (p. 27)
   - 3.11 RECALL: Recall the model (p. 27)
   - 3.12 LIST: Print the data (p. 27)
   - 3.13 AGAIN: Start over with a different data file (p. 27)
   - 3.14 MESSAGE: Information on changes to IVAN (p. 27)
   - 3.15 MISSING: Missing value estimates (p. 28)
   - 3.16 $COMMENT (p. 28)
   - 3.17 HELP (p. 28)
   - 3.18 END: Stop the program (p. 28)
4. Appendices
   - 4.1 Unbalanced analyses (p. 29)
   - 4.2 Tukey's test for non-additivity (p. 31)
5. How to access the program (p. 33)

0. Introduction

IVAN is an interactive computer program designed for the analysis of factorial designs. The initial version has many limitations. Consequently, many features that will eventually be part of this program are not yet available. For now, IVAN is limited to the analysis of data with 7 or fewer classifications or factors (including replications), 30 or fewer levels per factor, with factors crossed or nested in any way for a total of 1024 or fewer observations. Unbalanced data sets, either with unequal numbers of replications, or unequal nesting, can be treated as missing value problems.

Data sets with missing values are permitted.

The program is still experimental and subject to change. Hence bugs may be found from time to time. Documentation of any bugs, e.g., computer output, and/or suggestions for improvements, would be sincerely appreciated. They should be sent to Dr. Sanford Weisberg, University of Minnesota, 352 Classroom Office Building, St. Paul, MN 55108, (612) 373-1068. Each time a major revision of IVAN is made, the version number on the program header will be changed. Whenever the version number increases by a digit to the left of the decimal point, a new user's manual will be available.

IVAN is intended to be an instructional program, and hence many of the "automatic" features of some programs are not available. However, the output from IVAN is sufficient to allow a thoughtful user to get most results with only pencil and paper or a pocket calculator and a book of statistical tables.

Most of the program was written by Kenneth Koehler. Important sections of the program were added by Christopher Bingham and Sanford Weisberg. The project is supported by grants from the University of Minnesota Educational Development program.

The program is written in FORTRAN and should run on CDC CYBER computers with a KRONOS operation system.

Technical manual. This guide is intended as a user's manual, and no attempt has been made to discuss the details of the computations, the algorithms used, etc. An accompanying manual to be called the IVAN Technical Manual is in preparation and should be available soon.

Disclaimer. The program has been extensively tested and checked for accuracy, and, to the best of our knowledge, is accurate. However, neither the University of Minnesota nor any of the authors claim any responsibility for any errors that do arise.

1. Data files, file types and program limitations

Data to be used by IVAN must be on a local KRONOS file with any name beginning with a letter and seven or less characters. Because the form of the data to be input to the program is fairly complicated, we present here a complete example on the formation of a data file.

Suppose that the problem to be analyzed has a data structure with three factors or classifications (in general, up to 7 are permitted), and suppose the first classification has 2 levels, the second has 3 levels and the third has 4 levels (in general, up to 30 levels are permitted). This would be the appropriate structure for the following problems: 2 by 3 factorial design with 4 replications or 4 randomized blocks; a 2 by 3 by 4 design without replication; or a split plot design with 2 levels for the whole plot factor, 3 for the subplots,and 4 replications or blocks (the above list of possible models is not exhaustive). In all of these cases, the data consists of 2 x 3 x 4 = 24 numbers to be analyzed by the computer. The numbers may be uniquely determined by a multidimensional array Y_{ijk}, where

i, the leftmost subscript, refers to the level of the first classification or factor

j, the middle subscript, refers to the level of the second classification or factor

k, the rightmost subscript, refers to the level of the last classification or factor.

For example, Y_{213} would be the observation with the leftmost classification at level 2, the second classification at level 1 and the rightmost classification at level 3. Each observation can be uniquely determined in this fashion.

IVAN requires that the data on the file be entered in one of two specific orders determined by the levels of the various classifications. In the first alternative, the data is ordered with the levels of the

leftmost factor changing fastest, the 2nd factor changing 2nd fastest,..., the rightmost changing slowest. The resulting order for the data is given in column A of Table 1 for the 2 by 3 by 4 example. This ordering for the data is called left to right since the subscripts change from left to right.

Alternatively, the data may be in order with the last subscript changing fastest, the next to last changing second fastest,..., the left most changing slowest. For the example, this ordering is shown in column B of Table 1. This ordering is called dictionary order or right to left because, if we think of the subscripts as spelling a "word" (where in the example a word consists of three digits), then the words are in the order that one would find them in a dictionary -- the "smallest" word (1 1 1) first, and the "largest" word (2 3 4) last.

Table 1. Order of 2 x 3 x 4 data structure for (A) first subscript changing fastest and (B) last subscript changing fastest (dictionary order). The entries in the table are the subscripts corresponding to observations; e.g., 1 2 1 means Y_{121}.

(A) Left to Right			(B) Right to Left (dictionary order)		
i	j	k	i	j	k
1	1	1	1	1	1
2	1	1	1	1	2
1	2	1	1	1	3
2	2	1	1	1	4
1	3	1	1	2	1
2	3	1	1	2	2
1	1	2	1	2	3
2	1	2	1	2	4
1	2	2	1	3	1
2	2	2	1	3	2
1	3	2	1	3	3
2	3	2	1	3	4
1	1	3	2	1	1
2	1	3	2	1	2
1	2	3	2	1	3
2	2	3	2	1	4
1	3	3	2	2	1
2	3	3	2	2	2
1	1	4	2	2	3
2	1	4	2	2	4
1	2	4	2	3	1
2	2	4	2	3	2
1	3	4	2	3	3
2	3	4	2	3	4

Input files must have the data in either left to right or dictionary order. There is however, some flexibility in the layout of the file itself. The simplest type of file will consist of only the data, with no extraneous information, e.g. line numbers, labels or other variables. For this type of file, the user is restricted to having at most 25 observations per line, with a minimum of one observation per line. The number of observations on each line need not be the same.

As a slight modification of this, files may be written as above, but with the addition of line numbers, and no other extra information. The first number on each line is considered a line number and is ignored. In this case, the user is limited to no more than 24 observations (in addition to the line numbers) per line.

Finally, the user may have the data relevant for an analysis occuring in a single column of a data file which may contain other columns. This would be useful if, for example, the user has several responses to be analyzed with the same design. This will also allow eventual generalization of the program to analysis of covariance. For this option, the data must always be in the same column (e.g. always the third item on the line), no more than 25 columns per line, and with each line of the file corresponding to one observation. The remaining information on each line may be alphabetic or numeric. Thus, if desired, labels can appear on the file and processing of the data is still possible.

To summarize, there are currently 6 types of input files that can be successfully read by IVAN. These can be displayed in a two way table of orderings (left to right, dictionary) versus file format (data only; data and line numbers; read a specific column):

File format	Ordering (left to right)	Dictionary (right to left)
Data only, no line nos.	Type 1	Type 4
Data plus line numbers	Type 2	Type 5
Data in one column only	Type 3	Type 6

When using IVAN, the user is asked: "Name your data file and input type". The name of the data file is the name you have given it. The input type is shown in the above table. For example, suppose that local file DATAFIL contains the data for the 2 by 3 by 4 structure discussed so far, and the data is in left to right order with line numbers. In response to the initial question, the user could enter DATAFIL TYPE 2. (Note: In the type specification, the use of the word "type" is optional, so the following three are equivilent for specifying a type: TYPE2, TYPE 2, 2).

Missing data: Data to be used in IVAN must be entered in a completely balanced structure. Any missing values in the data must be indicated by a place holder. The place holders are the letter M or the symbol ? (question mark). Note that use of this option will permit the analysis of an unbalanced data set by making it balanced by filling in place holders.

Limitations and restrictions: The program is limited to problems with no more than 7 classifications (or 6 plus replications), with at most 30 levels per classification. However, in any event the total number of observations may not exceed 1024.

2. Model Specification

A model in IVAN is a symbolic statement of the equation to be fit to the data. The symbolism used is analogous to the usual symbolism used in describing mathematical models in analysis of variance.

Example 1 The mathematical model for the 2-way design on page 300 of Snedecor and Cochran is $Y_{ij} = \mu + \alpha_i + \beta_j + e_{ij}$, where μ represents the grand mean, α_i represents the i-th treatment effect, β_j represents the j-th replication effect of the experiment, and e_{ij} represents the residual for point Y_{ij}. In IVAN the model may be written in one of the following two forms:

(1) $$Y(IJ) = T(I) + R(J) + E(IJ).$$

(2) $$TR(IJ) = T(I) + R(J) + E(IJ).$$

Note that the term corresponding to μ does not appear in either (1) or (2), It could be included optionally (see page 8) but all other terms correspond to terms in the mathematical model (IVAN assumes that the grand mean is present in all models so the grand mean term does not have to be specified). The letters T and R indicate which factors are involved in each term, and are called factor symbols. In this example, T corresponds to the treatment effect and R corresponds to the replicate effect. The letter E always indicates an error term. The letters enclosed in parentheses, I and J, correspond to the subscripts in the mathematical model.

The models given by (1) and (2) differ in only one feature. The model given by (2) has all factor symbols used in the model listed in the term on the left of the equal sign. It is called a FORM-2 model. The model given by (1) uses the letter Y instead of a list of factor symbols in the term on the left of the equal sign. Models of that form are called FORM-1 models. Rules for each model form and examples of each are given below. It will be seen that FORM-2 models have an important shorthand option that FORM-1 models lack.

Rules for FORM-1 models:

1. The term left of the equal sign must have the letter Y in front of the first parenthesis.

2. The term left of the equal sign must have the same number of subscripts as there are subscripts for the data. Any letter may be used for a subscript, but each subscript must be represented by a unique letter. The model given in (1) has two subscripts I and J. The ordering of the subscripts must correspond to the ordering of the data on the data file.

3. All terms in the mathematical model must appear to the right of the equal sign (the overall mean is automatically added to all models, and therefore may be omitted). In example 1, the mathematical model is

$Y_{ij} = \mu + \alpha_i + \beta_j + e_{ij}$.

The IVAN model was given by

(1) Y(IJ) = T(I) + R(J) + E(IJ).

Here T(I) represents the term α_i, R(J) represents β_j, and E(IJ) represents e_{ij}. The overall mean, given by μ, is not specified. The IVAN model

(3) Y(IJ) = U + T(I) + R(J) + E(IJ).

is equivalent to the model given in (1). In (3) the overall mean is represented by the term U.

4. If the subscript I represents the levels of factor A and the subscript J represents the levels of factor B, then the interaction term must be written as AB(IJ) or BA(JI). Each subscript must occupy the same position among the subscripts as its corresponding factor symbol occupies among the factor symbols.

5. Any letter may be used as a factor symbol except for the letter E, which is always used to specify error terms. In split plot designs with several error terms, E is used to specify each error term (see example 3 below).

6. All terms must be separated by + signs.

7. Terms are fit in the order in which they appear in the model.

8. A model may be entered on several lines. However, ALL MODELS MUST END WITH A PERIOD.

Further examples of FORM-1 models are:

Example 2 For the two factor experiment with replication on page 347 of Snedecor and Cochran, the FORM-1 IVAN model is

(4) $$Y(IJK) = L(I) + S(J) + LS(IJ) + E(IJK) .$$

In this model, the factor symbol L represents the level of protein, and the symbol S represents the source of protein. Note that three subscripts are needed although there are only two factor symbols. The third subscript, K, identifies the K-th replicate. Therefore the term E(IJK) represents the residual effect of the K-th replicate for levels I and J of factors L and S, respectively.

Example 3 The split plot design on page 371 of Snedecor and Cochran can be specified by

(5) $$Y(IJK) = B(I) + V(J) + E(IJ) + D(K) + VD(JK) + E(IJK).$$

Note that the symbol E is used to designate both the whole plot error term, E(IJ), and the subplot error term, E(IJK).

Example 4 The hierarchal classification on page 286 of Snedecor and Cochran is specified by the Form-1 model.

(6) $$Y(IJK) = P(I) + L(IJ) + D(IJK).$$

In the experiment, 4 plants were selected at random. Then 3 leaves were randomly slected from each plant. Finally, two determinations of calcium content were made on each leaf. So we have determinations (D) nested within leaves (L) which are nested within plants (P). Then the term L(IJ) represents the effect of the J-th leaf from the I-th plant. Note that there is no error term.

Rules and examples will now be given for FORM-2 models. An example of a FORM-2 model was given by

(2) $\quad TR(IJ) = T(I) + R(J) + E(IJ)$

in example 1. Note that all factors symbols used in (2) are listed in the term on the left of the equal sign.

Rules for FORM-2 models:

1. The term on the left of the equal sign must have as many subscripts as there are subscripts in the data, and their ordering must correspond to the ordering of the data on the data file.

2. All factor symbols used in the model must appear before the first parenthesis in the term left of the equal sign. In (2), both factor symbols, T and R, appear in the term left of the equal sign. The factor symbols must appear in the same order as their corresponding subscripts. For example,

(7) $\quad RT(IJ) = T(I) + R(J) + E(IJ).$

would be an incorrect specification of the model for example 1, since on the right of the equal sign I is used as a subscript for the factor T, but T is paired with the subscript J on the left of the equal sign.

3. Keyword ALL. All terms in the mathematical model must be specified on the right side of the equal sign. This may be done by following rule 3 for FORM-1 models. However, there is a shorthand option available for FORM-2 models. By using the keyword ALL in place of subscripts, a single term can be made to represent several terms. For example,

AB(ALL2) is equivalent to A(I) + B(J) + AB(IJ).

AB(ALL1) is equivalent to A(I) + B(J).

Therefore the model given by (2) may be written as

(8) $\quad TR(IJ) = TR(ALL1) + E(IJ).$

In general, if k is an integer and k is less than or equal to the number of factor symbols in the term, then using ALLk in place of subscripts results

in the computation of all terms involving the factor symbols of order k or less. Further examples are given after the list of rules. The keyword ALL may not be used for the subscripts of error terms.

4. Rule 4 for FORM-1 models does not apply to FORM-2 models.

5. Rules 5, 6, 7, and 8 for FORM-1 models do apply to FORM-2 models.

Examples 2, 3, and 4, will now be used to illustrate FORM-2 models.

Example 2 The FORM-1 model given by (4) can be written in several ways as a FORM-2 model. The models

(9) LS(IJK) = LS(ALL2) + E(IJK).

(10) LSE(IJK) = LS(ALL2) + E(IJK).

(11) LS(IJK) = L(I) + S(J) + LS(IJ) + E(IJK).

are all equivalent to (4). Note that there may be fewer factor symbols than subscripts in the term left of the equal sign when the experiment has replicates. But, as in (10), the letter E may be used as a place keeper in the term to the left of the equal sign.

Example 3 For the split plot design, the FORM-1 model given by (5) can also be written in several ways as a FORM-2 model. The models

(12) BVD(IJK) = BV(ALL1) + E(IJ) + VD(ALL2) + E(IJK).

(13) BVD(IJK) = B(I) + E(IJ) + VD(ALL2) + E(IJK).

(14) BVD(IJK) = B(I) + V(J) + E(IJ) + D(K) + VD(JK) + E(IJK).

are all equivalent to (5). Note that both errors had to be explicitly given in (12), (13), and (14).

Example 4 The hierarchal model given by (6) can be written in only one way as a FORM-2 model. The model

(15) PLD(IJK) = P(I) + L(IJ) + D(IJK).

is equivalent to (6). It is not equivalent to

(16) PLD(IJK) = PLD(ALL3).

because PLD(ALL3) is equivalent to P(I) + L(J) + PL(IJ) +

D(K) + PP(IK) + LD(JK) + PLD(IJK).

The keywork ALL may only be used in place of subscripts for completely crossed factors. It cannot be used for nested factors.

After the user specifies the model, the computer prints out the model it has read. The computer numbers each error term. For example 3, if the user entered

(13) BVD(IJK) = BV(ALL1) + E(IJ) + VD(ALL2) + E(IJK).

the computer would print

YOUR MODEL IS

Y(IJK) = B(I) + V(J) + E1(IJ) + D(K) + VD(JK) + E2(IJK).

The whole plot error is labeled E1(IJ) and the subplot error is labeled E2(IJK). The numerical labels are used to distinguish the error terms in the ANOVA table and also in the specification of the RESIDUAL command (see sections 3.2 and 3.6).

If the user wishes to change the model from the one originally entered, the command MODEL can be used repeatedly.

3. COMMANDS

This section contains detailed descriptions of all the commands available in IVAN. Usage of most of the commands is illustrated using two examples taken from Snedecor and Cochran's Statistical Methods. Example 1, from page 300, gives the results from a 5 by 5 randomized block experiment, with rows as treatments and columns as blocks. The recorded values are the number of failures out of 100 planted soybean seeds. The data file is listed below:

```
 NULL      PROGRAM    SC300

00100 8 10 12 13 11
00110 2 6 7 11 5
00120 4 10 9 8 10
00130 3 5 9 10 6
00140 9 7 5 5 3
```

This is a type 5 data file, since it has line numbers, and we want varieties to be the first factor, but as entered (row-wise) the block factor is changing fastest.

Example 2, which is somewhat more complicated, is given on page 371 of Snedecor and Cochran. This is a split plot design carried out in randomized blocks, with varieties as a whole plot factor and dates of cutting as the sub-plot factor. The data file is given below. This is a type 1 file, with blocks (6 levels) as factor 1, varieties (3 levels) factor 2, and dates (4 levels) as the last factor. The data in the file is not in the same order as in Snedecor and Cochran.

```
LIST,F=SC371
2.17 1.88 1.62 2.34 1.58 1.66
2.33 2.01 1.70 1.78 1.42 1.35
1.75 1.95 2.13 1.78 1.31 1.30
1.58 1.26 1.22 1.59 1.25 0.94
1.38 1.30 1.85 1.09 1.13 1.06
1.52 1.47 1.80 1.37 1.01 1.31
2.29 1.60 1.67 1.91 1.39 1.12
1.86 1.70 1.81 1.54 1.67 0.88
1.55 1.61 1.82 1.56 1.23 1.13
2.23 2.01 1.82 2.10 1.66 1.10
2.27 1.81 2.01 1.40 1.31 1.06
1.56 1.72 1.99 1.55 1.51 1.33
```

Throughout the following sections, we shall refer to these two examples. (User input is underlined; computer response is not). However, no attempt at a complete analysis will be made. The user may access the data files and reproduce the examples. On MERITSS at the University of Minnesota, the files are stored as SC300 and SC371 on user number 2051999.

General comments. Any command may be abbreviated by its first 4 letters. Generally, any command that requires a complicated specification will have a help feature (e.g., POLY HELP), which will provide the novice (or forgetful) user information on command details.

3.1 MODEL. With this command, the user may specify a new model. When the program begins, the user is forced initially to specify a model, and, in the course of analysis, it may be convenient to change the model. The form of the model is discussed in detail in section 2.

For example 1, an appropriate model is given below:

```
X,DO,IVAN
    I V A N  -  VERSION 1.0
ENTER FILE NAME AND DATA INPUT TYPE OR 'HELP'.
? SC300 TYPE5
NUMBER OF OBSERVATIONS =  25
ENTER NUMBER OF LEVELS FOR EACH SUBSCRIPT, FROM LEFT TO RIGHT.
? 5 5
ENTER MODEL
? TB(IJ)=TB(ALL1)+E(IJ).
 YOUR MODEL IS
Y(IJ)=T(I)+B(J)+E1(IJ)
```

For the split plot example, the appropriate model is

```
X,DO,IVAN
    I V A N  -  VERSION 1.0
ENTER FILE NAME AND DATA INPUT TYPE OR 'HELP'.
? SC371 TYPE 1
NUMBER OF OBSERVATIONS =  72
ENTER NUMBER OF LEVELS FOR EACH SUBSCRIPT, FROM LEFT TO RIGHT.
? 6 3 4
ENTER MODEL
? BVD(IJK)=BV(ALL1)+E(IJ)+D(K)+VD(IK)+E(IJK).
 YOUR MODEL IS
Y(IJK)=B(I)+V(J)+E1(IJ)+D(K)+VD(IK)+E2(IJK)
```

Note that in this model the term BD(IK) is missing as this term is usually pooled with sub-plot error.

3.2 ANOVA. This is the basic command in IVAN. The output from this command is the overall analysis of variance table for the model specified. The terms are fit in the same order in which they appear in the model. The correction for the mean is also given. Each line in the table is numbered; the line numbers are used to obtain F-tests, as described in Section 3.3.

For example 1, the ANOVA command is given by

```
NEXT? ANOVA
                ANALYSIS OF VARIANCE
LINE NO.  TERM        DF    SUM OF SQUARES    MEAN SQUARE
   1      T           4        83.840           20.960
   2      B           4        49.840           12.460
   3      ERROR-1    16        86.560           5.4100
-------------------------------------------------------------
          TOTAL      24        220.24
          GRAND MEAN  1        1413.8
```

For example 2, it is

```
NEXT? ANOVA
                ANALYSIS OF VARIANCE
LINE NO.  TERM        DF    SUM OF SQUARES    MEAN SQUARE
   1      B           5        4.1498           .82996
   2      V           2        .17802           8.90097E-02
   3      ERROR-1    10        1.3623           .13623
   4      D           3        1.9625           .65416
   5      VD          6        .21056           3.50931E-02
   6      ERROR-2    45        1.2585           2.79677E-02
-------------------------------------------------------------
          TOTAL      71        9.1218
          GRAND MEAN  1        183.58
```

Missing values. If missing values occur in the data, either specified by an M or ? in the data file, or by the CHANGE command, the analysis of variance given by the ANOVA command is computed after filling in the missing values via least squares and proceeding as if the estimates were observed values (except that the degrees of freedom for error is appropriately reduced). This technique

has the following properties: (1) the error mean square is an unbiased estimate of the error variance, (2) the sums of squares for effects and interactions are slightly too large, correlated, and are not distributed as multiples of chi-square. Hence, the usual F-statistics will be slightly too large. However, if the number of missing values is small, the tests will be approximately correct.

A technique for obtaining "correct" F-tests is described in Section 4.1.

2.3 FTEST. F tests for effects and interactions in the anova table can be performed using this command. The form of the command is

FTEST [numerator list]/denominator

where the numerator list gives the line numbers of the mean squares for numerators of one or more F tests (separated by commas), and the denominator is the common denominator of all the F-tests (the slash (/) is mandatory). For Example 1, the F tests are obtained in this way:

```
      NEXT? FTEST 1,2/3
F(  4, 16) =    3.874
F(  4, 16) =    2.303
```

(Note: the second F test, for block effects, is probably not of interest in most contexts.) In the split plot experiment (Example 2), with two error terms, two FTEST commands are required to get all the desired test statistics.

```
      NEXT? FTEST 1,2/3
F(  5, 10) =    6.092
F(  2, 10) =    .6534

      NEXT? FTEST 4,5/6
F(  3, 45) =    23.39
F(  6, 45) =    1.255
```

In addition to the simple F-test illustrated above, it is possible to do tests with either the numerator or the denominator based on a pooled sum of squares, e.g. FTEST 1+2/3+4 would compute the pooled means squares based on lines 1 and 2, and lines 3 and 4 of an anova table and compute their ratio. Any number of terms may be pooled, and any number of tests may be done on one FTEST command (as long as they all have the same denominator). Thus, an FTEST command might look like: FTEST 1,1+2,1+2+3/4+5+6+7, which would produce 3 different F-tests with the same denominator.

The user may also enter FTEST HELP for a description of this command.

3.4 CONTRASTS. In factorial experiments with factors at more than two levels, it is rarely the case that the summary of the data given by the anova table is adequate. Usually, the analyst will be interested in examining certain contrasts (that is, linear combinations) among the means for various treatment combinations. This can be accomplished in IVAN through the use of two commands. The first of these, described here, is the CONTRAST command, allowing computation of arbitrary linear contrasts among means. The second, the POLY command (Section 3.5) is designed for use with quantitative factors for the computation of polynomial contrasts in the means.

In its simplest form,the CONTRAST command looks like:

CONTRAST [Factor label] [no. of contrasts to be computed]

For example, in example 1, a contrast of interest would be to compare the check condition (condition 1 of T) versus the average of the rest. Recall that a contrast is defined to be a linear combination of treatment means, e.g. Contrast = $\Sigma\lambda_i \times (i^{th}$ trt. mean), where, by definition, $\Sigma\lambda_i = 0$. The sample estimate of the contrast is just $L = \Sigma\lambda_i \times (i^{th}$ sample mean), and the associated sum

of squares is given by $nL^2/\Sigma\lambda_i^2$, where n is the number of observations that go into each mean (see page 308, Snedecor and Cochran). The number of contrasts may be any positive number; contrasts need not be orthogonal.

Thus, for the above example, the following exchange would occur:

```
        NEXT? CONTRASTS T 1
ENTER   5 COEFFICIENTS FOR CONTRAST NUMBER   1
? 4 -1 -1 -1 -1
LABEL   DF     CONTRAST      SS-CONTRAST     WEIGHTS
  T1     1       16.400          67.240       4.0000
```

In the output, the contrasts are each given unique labels. The column labeled "contrast" gives L as defined in the last paragraph, and the column "weights" gives $\Sigma\lambda_i^2/n$. Note that the user was asked to input the values of the λ's for the contrasts. The user should only input as many values as requested and recall that $\Sigma\lambda_i = 0$. If several contrasts had been specified, the user would have been asked to input several sets of λ's, one set at a time.

More complicated specifications. The CONTRAST command is capable of handling more complicated requests (although the above short description will serve the purposes of many users). The first complication is that the factor label for computation of the contrasts may have more than one letter; however, the total number of treatment combinations in the specified combination cannot be greater than the square of the maximum number of levels for any one factor; thus, contrasts among two factor means are always permitted, but contrasts among three factor means may not be possible, depending on the size of the problem. For example, in the 5 by 5 problem, specifying CONTRASTS TB 2 would be equivalent to asking for two contrasts in the 25 TB combinations, here each TB mean is just an original data point.

Keyword FOR. The final option for the CONTRAST command is the use

of the keyword FOR. A typical command would be

CONTRAST T 1 FOR TB

This would cause the computation of the specified contrast in the T means; however, computation would be done separately for each level of B. Thus, the term to the right of FOR must contain all the letters in the term to the left of the FOR. For the first example, the output for this command is given as follows:

```
        NEXT? CONTRASTS T 1 FOR TB
ENTER   5 COEFFICIENTS FOR CONTRAST NUMBER   1
? 4 -1 -1 -1 -1
```

LABEL	DF	CONTRAST	SS-CONTRASTS	WEIGHTS	B
T1	1	14.000	9.8000	20.000	1
T1	1	12.000	7.2000	20.000	2
T1	1	18.000	16.200	20.000	3
T1	1	18.000	16.200	20.000	4
T1	1	20.000	20.000	20.000	5

One would hope that, at least in this case, the value of the contrast would be approximately the same in each block, as it is in this case. Note that the 5 contrasts printed above, while orthogonal, do not all lie in the TB interaction space. If one adds the sum of squares for these five contrasts, one gets the sum of squares for the T1 main effect (1 df) plus the 4 df for the T1 × B interaction. Since the sum of squares for T1 was already found to be 67.240, the 4 df T1 × B interaction is equal to 9.8 + 7.2 + 16.2 + 16.2 + 20.0 - 67.240 = 2.16, with 4 df.

In sum, the form of the CONTRAST command is given by

CONTRASTS [factor label] [number of contrasts] FOR [factor label].

The user may enter CONTRAST HELP to obtain the above information.

Missing values. If there are missing values on the file, the sums of squares computed by this subroutine are only approximately correct, since they use the

"filled in" data. Currently, there is no way in IVAN to get the "correct" sums of squares. In most instances, however, if the number of missing values is small, the results produced by the program should be nearly correct.

3.5 POLYNOMIAL CONTRASTS. The form of this command is slightly more complicated than the form of most commands in IVAN. The general form is

POLY [factor letters] [keywords]

On any given POLY command, one or more of the keywords (described below) may be included. If a keyword is missing, a default value is assumed.

The POLY command results in the computation of polynomial contrasts for the main effects of factors specified as quantitative, as well as the associated sum of squares, F-tests and estimates of slope parameters.

For discussion of polynomial contrasts, the reader is referred to Snedecor and Cochran, Statistical Methods, pages 349-58 and 460-64.

The following is a description of the specification of a POLY command. If no specification is made, the computer is programmed to provide the user with prompts to allow for input of the necessary information.

a. The user must specify the letter names of the factors that are quantitative. For example, suppose we have the model Y(IJK) = A(I) + B(J) + AB(IJ) + E(IJK), that is, a two-way design with interaction and replication. The command POLY A would specify that A is a quantitative factor for which we wish to compute polynomial contrasts. If both A and B were quantitative, the appropriate command would be POLY AB.

b. Spacing Keyword. In most experimental designs, the levels of a quantitative factor are equally spaced. If this is in fact the case, then the keyword EQUAL could appear on the POLY command. For example, if factor A was the amount of a chemical additive (0g, 10g, 20g, 30g) the

command POLY A EQUAL could be used. If factor B was percent concentration of a second chemical (0%, 1%, 5%, 10%), then for factor B the levels are not equally spaced. Here the appropriate command is POLY B UNEQUAL. In this case, the user will then be asked to input the levels of the factor, which, for factor B, are 0, 1, 5, 10.

If both A and B appear on the same POLY command and B is unequally spaced, the appropriate form would be POLY A B UNEQUAL, even though the spacing for A is, in fact, equal. When asked to input the levels of factor A, simply input a set of equally spaced levels, e.g., 0 10 20 30.

If neither EQUAL nor UNEQUAL occur on the command card, the default EQUAL is assumed.

c. Keyword DEGREE. The degree of the polynomials to be fit is specified by the DEGREE keyword. For example, the command POLY A B DEGREE 2 would result in linear and quadratic polynomials being fit to both A and B. The same degree will be fit to all factors on one POLY command.

If the keyword DEGREE does not occur on the POLY command, the default value is DEGREE 1.

Using all the defaults, the following four commands are equivalent:

```
POLY A B
POLY A B EQUAL
POLY A B DEGREE 1
POLY A B EQUAL DEGREE 1
```

d. Keyword FOR. The keyword FOR is used to designate the effects for which contrasts are to be computed. The command

```
POLY A B FOR A B AB
```

would result in the computation of linear polynomial contrasts for A, B, and the AB interaction. If B were not quantitative then

POLY A FOR A , AB

would compute linear polynomial contrasts for the A main effect and in the A x B interaction.

The specification

POLY A B FOR ALL1

is equivalent to

POLY A B FOR A B.

The default value for FOR is FOR ALL1.

The complete form of a POLY command, showing all defaults, is:

POLY A B EQUAL DEGREE 1 FOR ALL1.

As an example, we compute the linear and quadratic trends among dates of cutting for the split plot example. First, assume that the dates of cutting are equally spaced.

```
      NEXT? POLY D DEGREE2
MAIN EFFECT POLYNOMIAL CONTRAST FOR FACTOR D
    CONTRAST          DF      SS-CONTRAST       MS-CONTRAST
  D-LIN                1      1.17361E-03       1.17361E-03
  D-QUAD               1        1.3972            1.3972
  REMAINDER            1        .56406            .56406
```

In reality, the dates of cutting were not equally spaced, but can be considered to be at levels 0, 35, 54, 71. Using three unequally spaced levels, the linear and quadratic trends are again computed. The results don't differ much from the above.

```
      NEXT? POLY D UNEQUAL DRE__EGREE2
YOU HAVE UNEQUAL SPACINGS.  SPECIFY THE LEVELS FOR EACH
QUANTITATIVE FACTOR WHEN IT IS PRINTED BELOW.
 SPECIFY THE  4 LEVELS FOR FACTOR D
? 0 35 54 71
MAIN EFFECT POLYNOMIAL CONTRAST FOR FACTOR D
    CONTRAST          DF      SS-CONTRAST       MS-CONTRAST
  D-LIN                1      7.83813E-02       7.83813E-02
  D-QUAD               1        1.6457            1.6457
  REMAINDER            1        .23838            .23838
```

3.6 MEANS. Form of command: MEANS [specification]. This command causes all marginal means to be printed. Specifications are of three forms:

1. The specifications ALL1, ALL2, ALL3 result in printing all 1-way, 2-way, and 3-way means, respectively. (Note: No space is permitted between "ALL" and "1".)

2. The specifications I, IJ, IK, IJK would result in the printing of the following (assuming the model is of the form Y(IJK)=terms; see sec. 2):

I - one-way means, adding over J, K

IJ - 2-way means, adding over K

IK - 2-way means, adding over J

IJK - 3-way means, e.g. the original data

3. The specification GM would print the grand mean.

Several specifications may be made simultaneously, e.g.

MEANS GM ALL1 IJ

is permitted, and would give a useful summary for a two-way design with replication.

Enter MEANS HELP for help at the teletype.

Missing values. If there are missing values on the file, the output from MEANS gives the correct least squares estimates of the appropriate means.

Example 1:

```
NEXT? MEANS ALL1     (Example 1)

   INDEX              MEAN      SS ABOUT MEAN
 1 +                 10.80          14.80
 2 +                 6.200          42.80
 3 +                 8.200          24.80
 4 +                 6.600          33.20
 5 +                 5.800          20.80
    OBS. PER MEAN        =          5
    SE(MEAN)             =      1.040
    SE(DIFF. OF 2 MEANS)=       1.471

   INDEX              MEAN      SS ABOUT MEAN
 + 1                 5.200          38.80
 + 2                 7.600          21.20
 + 3                 8.400          27.20
 + 4                 9.400          37.20
 + 5                 7.000          46.00
    OBS. PER MEAN        =          5
    SE(MEAN)             =      1.040
    SE(DIFF. OF 2 MEANS)=       1.471
```

3.7 RESIDUALS. The form of this command is RESIDUAL [error level]. The error level specification is needed only if more than one error term occurs in the model. A typical command would be RESIDUALS E2 . If only one error level is in the model, the specification of error level may be left off. Note that different models lead to different residuals. The output from this command for the first example is as follows:

```
        NEXT? RESIDUALS
RESIDUALS FOR ERROR LEVEL NUMBER   1
  INDEX          OBSERVED    FITTED VALUE    RESIDUAL      RES/S
I  J
1  1              8.000         8.480         -.4800        -.2064
2  1              2.000         3.880         -1.880        -.8083
3  1              4.000         5.880         -1.880        -.8083
4  1              3.000         4.280         -1.280        -.5503
5  1              9.000         3.480          5.520         2.373
1  2              10.00         10.88         -.8800        -.3783
2  2              6.000         6.280         -.2800        -.1204
3  2              10.00         8.280          1.720         .7395
4  2              5.000         6.680         -1.680        -.7223
5  2              7.000         5.880          1.120         .4815
1  3              12.00         11.68          .3200         .1376
2  3              7.000         7.080      -8.0000E-02   -3.4395E-02
3  3              9.000         9.080      -8.0000E-02   -3.4395E-02
4  3              9.000         7.480          1.520         .6535
5  3              5.000         6.680         -1.680        -.7223
1  4              13.00         12.68          .3200         .1376
2  4              11.00         8.080          2.920         1.255
3  4              8.000         10.08         -2.080        -.8943
4  4              10.00         8.480          1.520         .6535
5  4              5.000         7.680         -2.680        -1.152
1  5              11.00         10.28          .7200         .3096
2  5              5.000         5.680         -.6800        -.2924
3  5              10.00         7.680          2.320         .9974
4  5              6.000         6.080      -8.0000E-02   -3.4395E-02
5  5              3.000         5.280         -2.280        -.9802
     S=   2.326
```

The output from this command is the index of the observation, its observed value, fitted value, residual (= observed minus fit), standardized residual (= residual/(root mean square for error)). At the bottom of the output, the root mean square is printed.

The residuals can be examined to try to find discrepancies from the model, to try to find "bad values" or outliers in the data, or to suggest appropriate transformations of the measurements to a more useful scale. In the above table, the observation in the (5,1) cell seems to be discrepant, since the predicted value for this cell is 2.4 standard deviations below its observed value. It is likely that there is something odd about this observation, but that is hard to determine without consulting the original investigator. It is suggested that the researcher may want to reanalyze the data excluding the suspected point to see what the effect of this point is on the rest of the analysis. This is most easily accomplished using the CHANGE command, and is described in Section 3.9.

Output to a file. The residuals are automatically printed onto local file TAPE90 each time the residual command is invoked. The observed, fitted value, and residual for each point is printed, without heading or other information. If RESIDUAL is used several times, only the last call to it will be saved.

3.8 TRAN. This command transforms the input data. The untransformed data is then lost but can be restored by rereading the file (see AGAIN, sec. 3.10). The form of the command is

TRAN trf.

where trf is the name of the transformations to be used. The transformations available are:

ABS	Absolute value	
LOG	Logarithm, base 10	
LN	Logarithm, base e	
SQRT	Square root	
FT	Freeman-Tukey Deviate	$Y = \sqrt{Y} + \sqrt{Y + 1}$

ANGLIT $Y = \sin^{-1} \sqrt{Y}$

RECIP Y = 1/Y (not yet implemented)

The list of transformations is expected to increase in the future. Enter TRAN HELP for the current list.

3.9 CHANGE. This command is used to modify data in the data file, and can be very useful in exploratory data analysis. The command has two forms, CHANGE CELL and CHANGE DATA.

CHANGE CELL [cell subscripts] TO [newval]

results in the replacement of the value with the specified subscripts to NEWVAL, where NEWVAL is any number or a missing value symbol (M or ?).

CHANGE DATA [oldval] TO [newval]

results in all cells containing value oldval being changed to newval. Usually, either oldval or newval will be a missing value symbol (M or ?).

We illustrate the usefulness of this command with the data from example 1. As found under the RESIDUAL command, it is reasonable to analyze the data without the value in the 5,1 cell to find out the effect of this cell in the analysis. This is done by the following command:

```
      NEXT? CHANGE CELL 5 1 TO ?
CHANGE FROM   9.000      TO MISSING IN CELL   5 1
     NOTE  1 MISSING VALUES IN DATA SET.
```

To see the effect of this change, we would ordinarily repeat the analysis of the data set already done (e.g., get the analysis of variance table, recompute the contrasts, analyze the residuals, and cell means). With the exception of the anova table, we leave this as an exercise for the reader.

The modified anova table (with 5,1 assumed missing) is found by the ANOVA command to be

```
NEXT? ANOVA

                    ANALYSIS OF VARIANCE
LINE NO.  TERM        DF     SUM OF SQUARES     MEAN SQUARE
   1      T            4        125.41            31.353
   2      B            4        101.76            25.441
   3      ERROR-1     15         38.950            2.5967
-------------------------------------------------------------
          TOTAL       23        266.13
          GRAND MEAN   1       1287.0
```

The changes in this table are striking. The residual mean square has been cut by over 50% from 5.4 to 2.6, while the treatment mean square has increased by about 25%; the p-value of the F-test has changed from about .03 to less than .005 (from a table). Changes in means and contrasts should also be studied.

For further analysis of this data, see Section 4.1.

3.10 ERRORSS. This command prints degrees of freedom, sums of squares, mean squares and root mean square for each error term in the model.

The remaining commands on IVAN are primarily bookkeeping or informational, and do not initiate any computations. They can be used at any time.

3.11 RECALL. This command prints the model being used and the number of levels of each subscript. It is useful to users with terminals that do not have hard copy.

3.12 LIST (or DATA). This command causes the input data to be printed in left to right order (see Section 1 for details). The printing is done with five data points per line. For most purposes, the output from this command is very difficult to use. For small data sets, a more useful listing of the input data is available in the RESIDUAL command (Section 3.6). The output from this command is expected to improve substantially in the near future.

3.13 AGAIN. Start over with a different data file.

3.14 MESSAGE. The computer will print useful information on recent changes in IVAN. (This command may not be available on MIRJE.)

3.15 MISSING. At present this command prints the estimates of the missing cells and their locations.

3.16 $ COMMENT. Any line beginning with a $ is ignored by the computer. This command is useful for adding annotation to output. The blank following $ can be omitted if the first word typed in has no more than 9 characters.

3.17 HELP. The HELP command gives useful information if you don't have access to the manual. Many commands also have a "HELP option" by typing command HELP. (This command may not be available on MIRJE.)

3.18 END. Stop the program. Alternate forms are END or QUIT. The TELEX command STOP may also be used.

4. Appendices

4.1 Unbalanced analysis of variance.

Although IVAN is primarily designed for the analysis of balanced experiments, a facility does exist to enable the user to correctly analyze many unbalanced problems, obtaining "correct" tests of hypotheses, estimates of effects, and residuals. (The only exception to this is that the sums of squares attributable to contrasts, or polynomial contrasts, cannot be correctly computed with the current program). The basic technique is to treat any unbalanced data set as if it were really a balanced data set with missing values. This amounts to filling in a data set (with missing value symbols) so that the resulting "filled in" data is balanced. If the filling in requires no more than thirty missing value symbols, then the data set can be analyzed using IVAN. There are exceptions even to this rule. While the limitation to thirty missing values is caused by a limitation of space in the computer, the message "Pattern of missing data singular" is not. This means that, given the observed pattern of data, the effects included in the model are not all estimable. If this is the case, IVAN will fail to analyze the data because, with the given model, it cannot be analyzed; any program that would permit an analysis would be in error. The alternatives in these circumstances are to try a different model (one with fewer terms), or to change the data set.

In the following, we assume that a given unbalanced problem is to be analyzed, with IVAN, and that such an analysis is possible. As an example, we shall use the seemingly trivial running example in this manual, example 1, which is a 5 by 5 layout with the 5,1 cell deleted as a probable outlier. The problem is to obtain the "correct" estimation of test statistics, since parameters and residuals as given by the program are the correct (Gauss-Markov) estimates.

As stated in the description of the ANOVA command, the residual sum of squares, even with missing values, is the "correct" Gauss-Markov estimate of the

residual variance (under assumptions of the usual linear model), and it is orthogonal to all other sums of squares. The idea we shall use comes from these facts.

Suppose we wish to obtain the test of the hypothesis that there are no treatment effects; that is to say, we wish to test the null hypothesis NH: Y(IJ) = B(J) + E(IJ) against the alternative AH: Y(IJ) = T(I) + B(J) + E(IJ). We have already fit the alternative hypothesis model (in Section 3.9), and found that the residual sum of squares is 38.95 with 15 degrees of freedom. Now, suppose that we fit the null hypothesis model as below:

```
     NEXT? MODEL
ENTER YOUR MODEL
? Y(IJ)=B(J)+E(IJ).
```

We can obtain the residual sum of squares from this model as

```
     NEXT? ERRORSS

ERROR-1  DF=19  SS=   152.4        MS=   8.018        ROOT MS=   2.832
```

Now, the 19 d.f. SS for error here is the sum of squares for error from the alternative hypothesis plus the four degrees of freedom for fitting the treatment effects (strictly speaking, it is the sum of squares for treatments adjusting for blocks). Thus, if we subtract, 152.4 - 38.95, we will get the sum of squares for treatments, 113.4, which is not far from the value of 125.41 given by the approximate analysis of variance in Section 3.9 (if the number of missing values is larger, one should expect the discrepancy between the approximate and exact tests to grow).

The correct F test will always use the mean square error from the alternative hypothesis as a denominator. This F-test is given by

$$F = \frac{113.4/4}{2.597} = 10.916$$

By repeated application of this technique (e.g., fitting successive models with terms left out, and subtraction of residual sums of squares), virtually any analysis of variance may be obtained, giving the appropriate sums of squares adjusting for any other sums of squares, or ignoring any other set of sums of squares.

4.2 Tukey's test for non-additivity. Suppose we have a two-factor problem without replication for which we may write the model Y(IJ) = A(I) + B(I) + AB(IJ) + E(IJ). If AB(IJ)=0 for all I and J, we say that we have an additive model since the measured response is simply the sum of the effects of level I of A and level J of B, without any interactions. If we have a non-additive model, that is AB(IJ) $\neq$ 0 for some (I,J) pairs, then the AB(IJ) terms are completely confounded with the error terms E(IJ) (unless we have replication, which would tend to lessen this problem). To deal with the problem of analysis of potentially non-additive data, John Tukey suggested that a particular form for the interaction terms may be hypothesized. In particular, he makes the assumption that, if an interaction exists, it is of the form AB(IJ) = G × A(I) × B(J), that is, it is a multiplicative term. This form of interaction has great appeal, as it is an approximation to a wide class of possible interactions--see, for example, Snedecor and Cochran, pages 331-7 or H. Scheffé, 1958, The Analysis of Variance, page 130, for good discussion of Tukey's idea.

Since the above representation for the interaction requires the estimation of only one additional parameter (e.g. G above, the A(I) and B(J) are the main effects) a test for non-additivity can be obtained as a one degree of freedom contrast. In IVAN, this is done using the POLY command. We use Example 1, with factors T and B, as an example.

To do the computations, we want to take linear × linear contrast from the

TB interaction, with unequal spacings, where the levels of the spacings are the means of the various factor levels. Thus, the following is required for Tukey's test:

```
      NEXT? MEANS I J

   INDEX            MEAN       SS ABOUT MEAN
 1 +                10.80          14.80
 2 +                6.200          42.80
 3 +                8.200          24.80
 4 +                6.600          33.20
 5 +                5.800          20.80
    OBS. PER MEAN        =          5
    SE(MEAN)             =      1.040
    SE(DIFF. OF 2 MEANS)=       1.471

   INDEX            MEAN       SS ABOUT MEAN
 + 1                5.200          38.80
 + 2                7.600          21.20
 + 3                8.400          27.20
 + 4                9.400          37.20
 + 5                7.000          46.00
    OBS. PER MEAN        =          5
    SE(MEAN)             =      1.040
    SE(DIFF. OF 2 MEANS)=       1.471

      NEXT? POLY T B UNEQUAL FOR TB
YOU HAVE UNEQUAL SPACINGS.  SPECIFY THE LEVELS FOR EACH
QUANTITATIVE FACTOR WHEN IT IS PRINTED BELOW.
 SPECIFY THE  5 LEVELS FOR FACTOR T
? 10.8 6.2 8.2 6.6 5.8        [these are the means for factor T]
 SPECIFY THE  5 LEVELS FOR FACTOR B
? 5.2 7.6 8.4 9.4 7.0         [these are the means for factor B]
INTERACTION POLYNOMIAL CONTRASTS FOR FACTORS T BY B
     CONTRAST          DF      SS-CONTRAST       MS-CONTRAST
   T-LIN   * B-LIN      1        1.4957            1.4957
   REMAINDER           15        85.064            5.6710
```

From the output, the sum of squares for Tukey's contrast is 1.4957; the remainder mean square, 5.6710, estimates σ^2. Thus F = 1.4957/5.6710 is the F-test for non-additivity; in this case it is clearly not significant, indicating that non-additivity of this special multiplicative form is not present in the data. If F were significant, further analysis would be called for, especially residual analysis, and, possibly a data transformation will be in order.

5. How to Access the Program

The program is currently available on the MECC/MERITSS CYBER 72 computer at the University of Minnesota. It is accessed by typing

X, DO, IVAN

from any user number. The user should have a local file containing data before calling the program.

The following will then be printed by the computer:

I V A N VERSION 1.0

NAME YOUR DATA FILE AND TYPE OR ENTER HELP

?

The type of the data file is given in the table on page 5 of this manual. Please note that all user responses must be terminated with a carriage return.

On other computer systems, the mode of accessing the program will be different, and the user should enquire locally.